UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/827,739	04/06/2001	John H. Sandham		6702

36183 7590 12/21/2005

PAUL, HASTINGS, JANOFSKY & WALKER LLP
P.O. BOX 919092
SAN DIEGO, CA 92191-9092

EXAMINER	
PROCTOR, JASON SCOTT	

ART UNIT	PAPER NUMBER
2123	

DATE MAILED: 12/21/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/827,739	SANDHAM, JOHN H.	
	Examiner	Art Unit	
	Jason Proctor	2123	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *02 November 2005*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-12* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-12* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☒ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☒ All b)☐ Some * c)☐ None of:

1.☒ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114. Applicant's submission filed on 2 August 2005 has been entered.

Claims 1-13 were rejected in final Office Action dated 2 August 2005. Applicants' response has amended claims 1-5 and 9-12 and cancelled claim 13. Claims 1-12 are pending in this application.

Claims 1-12 have been rejected.

Claim Rejections – 35 USC § 101

The previous rejections of claims 5-12 under 35 U.S.C. § 101 have been withdrawn.

The Examiner has considered the numerous arguments presented regarding the previous rejections under 35 U.S.C. § 101. However, based on current guidance provided to the examining corps, because claims 1-4 recite a method comprising a single step of transforming data, the claims as written do not fall into the statutory categories set forth by 35 U.S.C. § 101.

35 U.S.C. § 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

1. Claims 1-4 are rejected under 35 U.S.C. § 101 because the claimed invention is directed to non-statutory subject matter.

Claims 1-4 recite methods that lack a useful, concrete, and tangible result and therefore present nonstatutory methods. Transforming numbers (memory access addresses) to produce "transformed" memory access addresses does not, in and of itself, achieve a useful, concrete, and tangible result. Such a method is merely an abstract mathematical process.

While physical transformation outside of a computer is evidence of a tangible result, computation within a computer is not a tangible result. There is always some form of physical transformation within a computer because a computer acts on signals and transforms them during its operation and changes the state its components during the execution of a process. Even though such a physical transformation occurs within a computer, such activity is not determinative of whether the process is statutory because such a transformation alone does not distinguish a statutory computer process from a nonstatutory computer process.

In claims 1-4, the method steps do not breathe life into the intended use recited by these claims' preambles because the single step of "transforming memory access addresses" is clearly insufficient to achieve that intended use. Merely transforming memory access addresses is not equivalent to and does not achieve "emulating a processor".

In order to overcome this rejection under 35 U.S.C. § 101, the Examiner respectfully suggests Applicants define the invention in claim language that breathes life into the intended use, such as a method step that positively recites how the "transformed memory access addresses" are employed in an active step of "emulating a processor". That is, the "transformed memory access addresses" should be described as a useful, concrete, and tangible result that will

be acted upon in a positively recited step to achieve "emulating a processor". Merely "using the transformed memory access addresses to emulate a processor" would be insufficient to convey this concept.

Claim Rejections – 35 USC § 112

The previous rejections under 35 U.S.C. § 112 have been withdrawn in light of Applicants' amendments.

2. Claims 2 and 10 are rejected under 35 U.S.C. § 112, first paragraph, as failing to comply with the enablement requirement. The claim(s) contains subject matter which was not described in the specification in such a way as to enable one skilled in the art to which it pertains, or with which it is most nearly connected, to make and/or use the invention. Claim 2 recites steps that are not operative for the stated purpose of "use in emulating a processor". Claim 10 recites a system that employs the method of claim 2 and fails to provide an "endian transformation system" as would be recognized by a person of ordinary skill in the art.

Making and/or using the invention of claims 2 and 10, which substantially recite creating an inverted memory image, where "the offset between addresses of any two bytes stored in memory is unaltered by the transformation, wherein said any two bytes are spaced apart in memory by a plurality of words, and the relative order of the addresses of said any two bytes stored in the memory is reversed by the transformation" would require numerous inventions and modifications to the "second type of processor," such as but not limited to a decrementing

program counter, a novel implementation of numerical data storage in memory, and a novel implementation of pointer arithmetic.

In every computer processor known to the Examiner, a program counter is automatically incremented to the next higher instruction address upon completion of an instruction. If the program instructions were reversed in memory, according to the method of claim 2, the computer processor would read the program instructions in the opposite order that they were intended. The automatic increment of a program counter is implemented in the hardware of a computer processor and not readily modified or adapted.

Similarly, both a big endian and little endian computer system share numerous properties related to memory addressing. It is only when storing data *within words* that they differ. Reversing the contents of an entire memory space, that is, *reversing words and their contents* would interfere with normal execution of both types of processors, especially regarding numerical data storage in memory and pointer arithmetic. The same arguments apply to the system of claim 10.

Claim Rejections - 35 USC § 102

The previous rejections of claims 2 and 10 under 35 U.S.C. § 102 have been withdrawn.

The following is a quotation of the appropriate paragraphs of 35 U.S.C. § 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an

> international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

3. Claims 1, 3-9, and 11-12 are rejected under 35 U.S.C. § 102(e) as being anticipated by US Patent No. 5,968,164 to Loen et al. (Loen).

Regarding claim 1, Loen discloses a method for adapting program code intended for a first processor to execute on a second processor which observes a different convention for ordering the significance of bytes within words wherein memory addresses for an entire address space are transformed to reflect the difference between big endian and little endian addressing conventions which result in a mirror image of the original bytes (column 6, line 20 – column 7, line 29; Figures 3A-3B).

In response, Applicants argue primarily that:

> Loen does not teach transforming memory access addresses of a plurality of words, as in claim 1 and claim 9.

The Examiner respectfully traverses this argument as follows.

As indicated in Applicants arguments, page 10:

> According to Loen, the most frequently accessed data words are repeatedly read and transformed, and transformed yet again if the words are changed and stored back into the memory address space. *See* Loen, col. 3, line 53 *et seq.*

Clearly Loen does teach transforming memory access addresses of a plurality of words. Loen anticipates the invention as claimed. The Examiner notes that Applicants have identified the invention of Loen as an example of the prior art. The Examiner respectfully suggests that Applicants define the invention in the claims with language that distinguishes the invention from the prior art.

Applicants' arguments have been fully considered but have been found unpersuasive.

Regarding claim 3, Loen et al. also discloses that byte strings stored successively by the processor aggregate in the same manner as if the code had been executed on a processor that is naturally biased toward the same endian convention as the code (column 6, line 20 – column 7, line 29; Figures 3A-3B). Loen discloses that *"The first step is a reflection which must be performed on the bytes comprising the data double word or fragment thereof (see FIG. 3a)."* (column 7, lines 5-7). FIG. 3a discloses that a "string of bytes" are store successively in accordance with the second endian format.

In response, Applicants argue primarily that:

> Loen does not fulfill the condition that, for a plurality of words, strings of bytes in the first endian format aggregate as recited.

The Examiner respectfully traverses this argument as follows.

The Examiner respectfully submits that Applicants' assessment of Loen on this point is inaccurate. As is well known in the art and described by the specification at page 1, a big endian system has its most significant byte addressed X while a little endian system has its least significant byte addressed X. *This is precisely the type of transformation disclosed by Loen* (column 6, lines 48-60, et cetera). That is, Loen is *expressly directed* toward the transformation of memory access addresses so that bytes are stored "in the same manner as the bytes would aggregate if the processor was of the [other] endian format and memory access addresses were not transformed."

Applicants' arguments appear to imply that a little endian computer system uses a memory addressing scheme that is *perfectly opposite* to that of a big endian system. This allegation is not supported by what is well known in the art [An example is given by Loen (column 5, lines 7-45)]. A little endian computer system uses a different method of addressing

the bytes *within words*, however the words are addressed equivalently in either system, i.e. as 0...n. Thus Applicants' insistence that the prior art must not only manipulate the bytes within words, *but also rearrange the words* in order to store bytes "in the same manner as the bytes would aggregate if the processor was of the [other] endian format and memory access addresses were not transformed" are not persuasive. These two concepts are at odds and define separate and distinct concepts.

Applicants' arguments have been fully considered but have been found unpersuasive.

Regarding claim 4, Loen et al. also discloses that the memory address B of word length L is transformed to the address A-B-L+S where A is the total number of bytes allocated to the program and S is the start address of the program (Figures 4A-4D). Figures 4A-4D of Loen et al. discloses an example where A=8, S=0, and L is 8, 16, 32, or 64 bit, however also teaches other values for L (column 7, lines 62 – column 8, line 46).

For example, FIG. 4B shows the recalculated offset to access a halfword (2 bytes) after reflecting the memory. In this case, the address being accessed is B=4. The length of the data being accessed is L=1. The new offset as shown in FIG. 4B is A-B-L+S = 8-4-2+0 = 2.

In response, Applicants argue primarily that:

> The two examples cited by the Examiner referring to Figures 4A and 4B of Loen require an improper interpretation of the claim language.
>
> Concerning the Figure 4A example, the condition recited in claim 4 is only achieved by Loen when the total memory allocated is A equals 8 bytes and the start address of the program is S equals 0 bytes. The transformation fails if the total memory allocated is any greater than 8 bytes, or where the program does not start at address 0. Any real program in a practical computer system operates on a far greater memory allocation than 8 bytes, and would not start at address 0.

The Examiner respectfully traverses this argument as follows.

Applicants' argument appears to misconstrue the Examiner's *example* as a statement of *requirements*. For Applicants' convenience, the Examiner will provide herein an example that directly refutes Applicants' allegation that "the condition recited in claim 4 is only achieved by Loen when the total memory allocated is A equals 8 bytes and the start address of the program is S equals 0 bytes."

Suppose for example that memory location 0 in FIG. 4A shows the relative start address of the program at memory location 1,000,000. That is, S=1,000,000.

In the top portion of FIG. 4A, relative memory location B=6 with length L=1 refers to data value "A1". Naturally, a person of ordinary skill in the art would recognize that the actual memory location referred to is 1,000,006, because the program's start address is S=1,000,000 as shown above.

In the bottom portion of FIG. 4A, the *transformed memory address* is relative memory location 1, as can be clearly seen in the figure referring to the same data value "A1". A person of ordinary skill in the art would recognize that the actual memory location referred to is 1,000,001, because the program's start address is S=1,000,000 as shown above. That is, A-B-L+S = 8-6-1+1,000,000 = 2-1+1,000,000 = 1+1,000,000 = 1,000,001. Memory location 1,000,001 corresponds to relative memory location 1, as indicated above. Additional examples for different values of B and/or S can be imagined.

This example directly refutes Applicants' allegation that "the condition recited in claim 4 is only achieved by Loen when the total memory allocated is A equals 8 bytes and the start address of the program is S equals 0 bytes."

Although Applicants allege that the examples cited in the previous office action "require an improper interpretation of the claim language," the Examiner can find no evidence to support this allegation. Further, it is unclear from Applicants' argument what interpretation of the claim language would be proper or how to distinguish that interpretation from the Examiner's supposed improper interpretation.

Applicants further argue that:

> Claim 4 recites transforming a plurality of memory access addresses relating to a plurality of words. Loen only transforms data word by word (i.e. a double data word or fragment thereof).

The Examiner respectfully traverses this argument as follows.

The Examiner respectfully submits that he does not follow Applicants' line of reasoning. Loen clearly discloses "transforming a plurality of memory access addresses relating to a plurality of words" in FIG. 4A-D, among other citations, which clearly show "transforming a plurality of memory access addresses relating to a plurality of words" (see also column 7, line 35 – column 8, line 18).

Applicants' arguments have been fully considered but have been found unpersuasive.

Regarding claim 5, Loen et al. discloses a process for translating a program code instruction for execution on a programmable machine using a corresponding predetermined convention of ordering the significance of bytes within words of the address space (column 6, line 20 – column 7, line 29; Figures 3A-3B) comprising:

transforming the referenced memory address with respect to a fixed block size of memory in the programmable machine so as to change the referenced address value by an

amount that is fixed for a given number of bytes being accessed in each word (Figures 4A-4D; column 7, line 62 – column 8, line 46);

including the changed address reference in the output instruction so that there is no extra operation required during execution of the instruction to accommodate the convention for ordering bytes within words used by said predetermined programmable machine (column 7, line 62 – column 8, line 46).

In response, Applicants argue primarily that:

Loen does not perform memory address transformation

and

Loen does not [include] the thus changed address reference in a compiled or translated output instruction so that there is no extra operation required during execution of the output instruction to accommodate the convention for ordering bytes within words used by said predetermined programmable machine.

The Examiner respectfully traverses this argument as follows.

As addressed above, the Examiner maintains that Loen performs memory address transformation (FIGS. 4A-D; column 7, line 35 – column 8, line 18).

Regarding Applicants' second point, the cited portion of Loen discloses "including the thus changed address reference in a translated output instruction" as required by the claim [*"To complete the processor fetch from memory, an address modification is performed on the address as originally presented by the software...Since these steps are hidden by PowerPC computer systems, a programmer cannot tell that the environment is anything other than little endian."* (column 7, line 62 – column 8, line 18)].

Regarding claim 6, Loen et al. discloses that the system is applied to program source code (column 7, lines 62 – column 8, line 46).

Regarding claim 7, Loen et al. discloses that a fixed block of memory is either big endian or little endian and therefore addressed from a predetermined one of its two ends depending upon the convention utilized for ordering the significance of bytes within the words (column 6, line 20 – column 7, line 29; Figures 3A-3B).

Regarding claim 8, Loen et al. discloses that the translation causes a fixed block of memory contents for a big-endian machine to be inverted to the mirror image of that for a little-endian machine (column 6, line 20 – column 7, line 29; Figures 3A-3B).

Claims 9 and 11-12 all recite "an endian transformation system" which performs the methods of claims 1 and 3-4, respectively. As the invention disclosed by Loen et al. is indeed a system that performs a method used to reject claims 1 and 3-4 (column 4, lines 49-55), claims 9 and 10-12 are rejected for the same reasons used to reject claims 1 and 3-4 above.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. § 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

The factual inquiries set forth in *Graham* **v.** *John Deere Co.*, 383 U.S. 1, 148 USPQ 459 (1966), that are applied for establishing a background for determining obviousness under 35 U.S.C. § 103(a) are summarized as follows:

1. Determining the scope and contents of the prior art.
2. Ascertaining the differences between the prior art and the claims at issue.
3. Resolving the level of ordinary skill in the pertinent art.
4. Considering objective evidence present in the application indicating obviousness or nonobviousness.

4. Claims 2 and 10 are rejected under 35 U.S.C. § 103(a) as being unpatentable over "Data Structures & Other Objects Using C++" by Michael Main and Walter Savitch (Savitch).

Regarding claim 2, Savitch discloses a method comprising transforming memory access addresses such that (a) the offset between addresses of any two bytes stored in memory is unaltered by the transformation, wherein said any two bytes are spaced apart in memory by a plurality of words, and (b) the relative order of the addresses of said any two bytes stored in the memory is reversed by the transformation (Chapter 7 Stacks; particularly "Programming Example: Reversing a Word", pages 310-311).

Savitch describes a method where a word (here referring to an English language word, using "NAT" as exemplary input) is manipulated to produce "TAN". As would be well known to a person of ordinary skill in the art, character data is typically stored as one character corresponding to one byte. Thus Savitch discloses a sequence of three bytes storing N, A, and T, respectively, being reversed such that they store T, A, and N. This transformation clearly meets requirements (a) and (b) as set forth in the claim. Savitch provides, in lesser detail, a longer example of the input data "ESIOTROT" being transformed to "TORTOISE" (page 310).

Savitch does not expressly disclose the intended use recited by the claim. However, a stack is a basic data structure that would be a well-known programming tool to a person of ordinary skill in the art. Therefore it would be obvious to a person of ordinary skill in the art at the time of Applicants' invention to use basic and well-known programming tools where appropriate. Further, Applicants admit that the intended use recited by the claim, specifically emulating a particular processor on another processor, is known in the art and not the object of invention, *per se* (specification, page 1).

Regarding claim 10, which recites a system for performing the method of claim 2, Savitch discloses concepts of computer programming (pages 306-307) and therefore implicitly discloses that a conventional computer system would perform the method. Claim 10 is therefore rejected for the same rationale given above regarding claim 2.

Conclusion

Art considered pertinent by the examiner but not applied has been cited on form PTO-892.

"Structured Computer Organization" by Andrew S. Tanenbaum provides a succinct example of how a big endian and little endian computer system store data (Section 2.2.3) and illustrates an example of transforming data from one endian to the other (page 60).

US Patent No. 5,907,865 provides a multiplexor which switches stored data from big endian or little endian to a common CPU-internal format while also aligning data reads for byte and halfword accesses in response to control signals (abstract).

US Patent No. 5,398,328 provides a method and apparatus for enabling a computer to run using either a Big Endian or Little Endian architecture. The conversion method and apparatus is implemented in software by scanning the instructions of the input for load and store instructions. The software modifies the instructions by taking the contents of the register and XORing the two least significant bytes of the byte address with a binary 3 (abstract).

US Patent No. 5,828,884 (PTO-892 part of paper number 20041110) provides a method for compiling a software program and executing the program on a data processing system which performs conversion between data formatted in differing endian formats, namely big-endian and little-endian formats, also known as byte swapping (abstract).

Consideration of the cited art is required prior to responding this Office Action. Please see 37 CFR 1.111(c).

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Jason Proctor whose telephone number is (571) 272-3713. The examiner can normally be reached on 8:30 am-4:30 pm M-F.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Leo Picard can be reached at (571) 272-3749. The fax phone number for the organization where this application or proceeding is assigned is (571) 273-8300.

Any inquiry of a general nature or relating to the status of this application should be directed to the TC 2100 Group receptionist: 571-272-2100. Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Jason Proctor
Examiner
Art Unit 2123

jsp

Paul L. Rodriguez 12/15/05
Primary Examiner
Art Unit 2125